FAKE NEWS DETECTION USING NLP

**Phase 5 submission document**

Phase 5:Project Documentation & submission

ABSTRACT:

 Fake news is information that is false or misleading but is reported as news. The tendency for people to spread false information is influenced by human behaviour; research indicates that people are drawn to unexpected fresh events and information, which increases brain activity. Additionally, it was found that motivated reasoning helps spread incorrect information. This ultimately encourages individuals to repost or disseminate deceptive content, which is frequently identified by click-bait and attention-grabbing names.

The proposed study uses machine learning and natural language processing approaches to identify false news specifically, false news items that come from unreliable sources. The dataset used here is ISOT dataset which contains the Real and Fake news collected from various sources. Web scraping is used here to extract the text from news website to collect the present news and is added into the dataset. Data pre-processing, feature extraction is applied on the data. It is followed by dimensionality reduction and classification using models such as Rocchio classification, Bagging classifier, Gradient Boosting classifier and Passive Aggressive classifier. To choose the best functioning model with an accurate prediction for fake news, we compared a number of algorithms.

INTRODUCTION:

Fake news is false or misleading information presented as news. The proposed study uses machine learning and natural language processing approaches to identify false news—specifically, false news items that come from unreliable sources.

Fake news and disinformation are ongoing problems that may be found all around us in biased software that amplifies just our viewpoints for a "better" and smoother user experience. Fake news and misinformation are becoming more of a problem as the internet and social media platforms become more main stream. A common goal of fake news is to harm someone or something's reputation or to profit through advertising. The propagation of these ideas may have been influenced by a variety of factors, but they all present humanity with the same underlying problem: a misunderstanding of what is real and what is false. This confusion could result in additional problems, such a medical emergency.

Satire websites or sensationalistic "click-bait" is where news site proprietors most frequently steal from fake. Satire websites frequently publish outlandish news parodies, and visitors to these websites are aware that they should not be taken seriously. Tabloids are what click-bait news articles resemble. Even though the actual tales are relatively mundane, their sensationalist and dramatic headlines entice people to click to learn more. These kinds of headlines draw readers to buzz feed and up worthy.

In this paper, we are focusing on the fake news detection in text media. Machine learning and deep learning techniques for fraud detection has been the subject of extensive study, most of which has concentrated on categorising online reviews and publicly accessible social media posts. Some of the drawbacks of the fake news are shift in public opinion, defamation, false perception and many more.

To overcome the drawbacks of the fake news, a model is created to distinguish the real news from the fake news. The proposed method uses the ISOT dataset. Web scraping is also done on 4 websites and the scraped data is further added into the dataset. The data undergoes data pre-processing, feature extraction, dimensionality reduction and finally the data is sent to the classification models i.e. Rocchio classification, Bagging classifier, Gradient boosting classifier and Passive Aggressive Classifier to train the model which is further used to detect the fake news.

The paper is organised as follows. The section covers the previous research on methods for identifying fake news. The proposed approach is described in the third section. Section four describes the evaluate

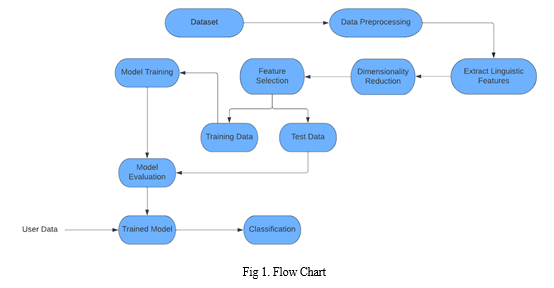
**II.  LITERATURE SURVEY**

There have been several previous works done to detect the fake news. Some of the previous works are given below. A fair amount of research has been done different aspects of this project domain.

1. According to "Fake News Detection Using Machine Learning Ensemble Methods" paper [1] the goal is to develop a system or model that can use historical data to forecast if a news report is fake or not. The dataset used here is ISOT dataset. The model used in this method is Random Forest Classifier. A large number of decision trees are built during the training phase of the random forests or random decision forests ensemble learning approach, which is used for classification, regression, and other tasks. The class that the majority of the trees chose is the output of the random forest for classification problems. Accuracy is one factor to consider when evaluating categorization models. The accuracy of the proposed solution is 90.64.
2. The author T. Dinesh. et. al. [2] has introduced a training model that can be used to predict the class or value of the target variable by learning simple decision rules developed from training data. The approach of classifying the fake political news manually requires more knowledge of the domain. In this research, the problem of classifying fake political news articles using machine learning models is discussed. The mean accuracy and standard deviation for the Decision Tree algorithm is 99.6990 and 0.10577. For Naive Bayes algorithm is 95.3870 and 0.00061. The accuracy of innovative fake news detection for political news detection using Decision Tree algorithms has better accuracy in comparison with Naive Bayes algorithms.
3. According to the author Kasra Majbouri. et. al. [3], the purpose is running K-Means clustering to see if the algorithm can successfully cluster the news into Real and Fake using just the words in the articles. The proposed method of choosing features and detecting fake news has four main steps. The first step is computing similarity between primary features in the fake news dataset. Then, features are clustered based on their similarities. Next, the final attributes of all clusters are selected to reduce the dataset dimensions. Finally, fake news is detected using the k-means approach. The accuracy of the K-means clustering algorithm in the detection of fake news is approximately 87%.
4. The goal of fake news identification, according to the author Monther Aldwahedi et al. [4], is to develop a method that users may use to identify and filter out websites that contain inaccurate and misleading information. A simple and carefully selected feature of the title and post to accurately identify fake posts. The experimental results showed a 99.4% accuracy using the logistic classifier.

**III. FAKE NEWS DETECTION USING NATURAL LANGUAGE PROCESSING**

Most text and documents contain many terms that are redundant for text classification, such as stop words, misspellings, slangs, and so on. Hence, data pre-processing has to be done before the data is sent to the classification models. After that, the dataset's dimensionality is decreased in order to save time and storage space. When the dimensions are reduced, it becomes easier to visualise. The data is then used to train classification models, which can be used to predict whether or not the presented data is fraudulent.



A. *Description of Dataset*

The dataset used in this paper is ISOT dataset. In this dataset, there are two types of articles: fake news and real news. The dataset was gathered from real-world sources, and true articles were retrieved via crawling articles from Reuters.com. The fake news articles came from a variety of sources. Politifact and Wikipedia were used to gather the fake news items. Although the majority of the articles in the collection are about politics and foreign events, they cover a wide range of topics. The dataset consists of two CSV files. True.csv is the first file, and it contains almost 12,600 reuter.com stories. Fake.csv, the second file, comprises about 12,600 items obtained from various fake news sites.

B. *Web Scraping*

Large volumes of data can be automatically gathered from websites via web scraping. The majority of this data is unstructured in HTML format and is transformed into structured data in a database or spreadsheet so that it can be used in multiple applications.

Here, web scraping is done on 4 websites to get the present news. It is further added into the dataset to detect the present news as fake or not and also to increase the efficiency of detecting the fake news.

C. *Text Cleaning and Pre-processing*

To prepare the text data for the model building we perform text pre-processing. It is the first step in the Natural Language Processing. Some of the pre-processing steps are:

1. *Tokenization:* Tokenization is the process of breaking down a stream of text into tokens, which can be words, phrases, symbols, or any other significant items. This step's major purpose is to extract individual words in a sentence. The tokenization is done on each text in the dataset.
2. *Stop Words:* Stop words are the commonly used words and are removed from the text as they do not add any value to the analysis. These phrases have little or no meaning. A list of terms that are regarded as stop words in the English language is included in the NLTK library. All the stop words from the texts are removed.
3. *Capitalization:* Sentences can have a combination of capital and lowercase letters. A written document is made up of multiple sentences. One of the method for reducing the issue space is to convert everything to lower case. This aligns all of the words in a document in the same location. Using the python function, all the words are converted to lower case.
4. *Stemming:* Stemming is the process of reducing the words to its root form by eliminating extraneous characters. PorterStemmer is one of the stemming model which is used here to convert the words into its root form.
5. *Lemmatization:* Text lemmatization is the process of removing a word's superfluous prefix or suffix and extracting the basic word. All the suffixes and prefixes from the words are removed to reduce space.

D. *Feature Extraction*

TF-IDF stands for Term Frequency-Inverse Document Frequency and it is a measure, used in the fields of information retrieval and machine learning that can quantify the importance or relevance of string representations in a document amongst a collection of documents. The Bag of Words technique, which is useful for text classification or for assisting a machine read words in numbers, is outperformed by the TF-IDF technique when it comes to understanding the meaning of sentences made out of words. Each feature's TF-IDF weights are computed and recorded in a matrix with columns denoting features and rows denoting sentence

E. *Dimensionality Reduction*

Dimensionality refers to how many input features, variables, or columns are present in a given dataset, while dimensionality reduction refers to the process of reducing these features. In many circumstances, a dataset has a significant number of input features, which complicates the process of predictive modelling.

In these situations, dimensionality reduction techniques must be used because it is highly challenging to visualise or forecast for the training dataset with a huge number of features. The curse of dimensionality, which is more commonly known, describes how adding more input features frequently makes a predictive modelling task more difficult to model. Since the TF-IDF matrix is a sparse matrix, Singular Value Decomposition is used for dimensionality reduction.

1. *Singular Value Decomposition:*Singular Value Decomposition is one of several techniques that can be used to reduce the dimensionality, i.e., the number of columns, of a dataset. A matrix's Singular Value Decomposition is a factorization of that matrix into three other matrices. Finding the ideal set of variables that can most accurately predict the result is the aim of SVD. During data pre-processing prior to text mining operations, SVD is used to find the underlying meaning of terms in various documents.

Mathematics behind SVD,

The SVD of mxn matrix is given by the formula,

A = UWVT

where

U: mxn matrix of the orthonormal eigenvectors of AAT

V: transpose of mxn matrix containing the orthonormal eigenvectors of ATA

W: a nxn diagonal matrix of the singular values which are the square roots of the eigen values of ATA

The matrix from TF-IDF is given as input to the TruncatedSVD. The columns i.e. features denotes the dimensions whereas the rows in the matrix denotes the points in the space. The dimensions of the matrix are reduced using TruncatedSVD.

F. *Classification Techniques*

On the basis of training data, the Classification algorithm is a Supervised Learning technique that is used to categorise new observations. The classification algorithms used in this paper is,

1. *Rocchio Classification:* A type of Rocchio relevant feedback is Rocchio classification. The centroid of the class of relevant documents is the average of the relevant documents, which corresponds to the most important component of the Rocchio vector in relevance feedback. Rocchio classification, which uses centroids to define the boundaries, is used to compute good class boundaries. Rocchio classification calculates the centroid for each class. When a new text data is given, it calculates the distance from each of the centroid and assigns the data point to the nearest centroid.
2. *Bagging:* When the goal is to reduce the variance of a decision tree classifier, bagging is utilised. The goal is to construct different subsets of data from a training sample that was picked at random and replaced. Their decision trees are trained with each group of data. As a result, we have a collection of various models. The average of all the forecasts from various trees is used which is more robust than a single decision tree classifier.
3. *Gradient Boosting:* A method for creating a collection of forecasts is called boosting. In order to reduce training errors, boosting is an ensemble learning technique that combines a number of weak learners into a strong learner. A random sample of data is chosen, fitted with a model, and then trained successively in boosting; each model attempts to make up for the shortcomings of the one before it. The weak rules from each classifier are joined during each iteration to create a single, powerful prediction rule. Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error.
4. *Passive Aggressive Classifier*: For large-scale learning, passive-aggressive algorithms are commonly used. It is one of the few 'online-learning algorithms'. In contrast to batch learning, where the full training dataset is used at once, online machine learning algorithms take the input data in a sequential order and update the machine learning model step by step. This is quite helpful when there is a lot of data and training the entire dataset is computationally difficult because of the size of the data. Since, the web scraping is used in this method, it adds the data to the dataset, and the size of the dataset becomes large which makes the Passive Aggressive Classifier model to work efficiently.

**ADVANTAGES:**

1.NLP algorithms can process large volumes of text data quickly, making it efficient for analyzing numerous news articles in real-time.

2.NLP models can work 24/7 without human intervention, allowing for continuous monitoring and detection of fake news.

3.NLP models can understand context, tone, and language nuances, making them effective in identifying subtle clues that may indicate fake news.

4.NLP models can be adapted to multiple languages, broadening the scope of fake news detection across different regions.

5.NLP models can learn from vast datasets, improving accuracy over time as they encounter new types of fake news.

6.NLP can rapidly process information, which is crucial in the digital age where news can spread quickly.

**APPLICATION:**

1. Detecting and mitigating the spread of fake news on platforms like Facebook, Twitter, and Instagram to maintain the credibility of information shared online.

2.Filtering and flagging potentially fake news articles before they are presented on news aggregator websites or apps.

3.Ensuring that user-generated content on websites and forums is free from fake news and misinformation.

4.Integrating NLP-based fact-checking tools within news websites and search engines to verify the accuracy of news articles.

5.Monitoring and debunking false or misleading claims made by political candidates during election campaigns.

**DESIGN THINKING AND PRESENTATION IN FORM OF DOCUMENT**

1. \*\*Data Collection\*\*: Gather a dataset of news articles with labels indicating whether they are real or fake. There are several publicly available datasets for this purpose, such as the Fake News Challenge dataset or fact-checking websites' data.

2. \*\*Preprocessing\*\*: Prepare the text data for analysis by performing various preprocessing steps like tokenization, stop-word removal, stemming/lemmatization, and lowercasing. These steps help in standardizing the text data and reducing noise.

3. \*\*Feature Extraction\*\*: Transform the text data into numerical features that machine learning models can work with. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec, GloVe, or BERT embeddings. These embeddings capture the semantic meaning of words and phrases.

4. \*\*Feature Engineering\*\*: Create additional features that might be informative for fake news detection. For example, you could include features like article length, the publication source, the publication date, or the number of hyperlinks.

5. \*\*Model Selection\*\*: Choose an appropriate machine learning model for classification. Common choices include:

- \*\*Multinomial Naive Bayes\*\*: A simple and efficient model for text classification.

- \*\*Logistic Regression\*\*: Often used for binary classification tasks.

- \*\*Random Forest\*\*: An ensemble model that can capture complex patterns.

- \*\*Support Vector Machine (SVM)\*\*: Effective for high-dimensional data.

- \*\*Deep Learning Models\*\*: Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can also be used for NLP tasks.

6. \*\*Training and Testing\*\*: Split the dataset into training and testing sets. Train the selected model on the training data and evaluate its performance on the testing data using metrics like accuracy, precision, recall, F1-score, and ROC AUC.

7. \*\*Tune Hyperparameters\*\*: Optimize the model's hyperparameters to achieve the best performance. Techniques like grid search or random search can help with this.

8. \*\*Ensemble Methods\*\*: Combine multiple models to improve accuracy and robustness. Ensemble methods like stacking, bagging, or boosting can be useful.

9. \*\*Real-Time Monitoring\*\*: Implement a real-time monitoring system that can analyze news articles as they are published. This might require continuous data collection and processing.

10. \*\*Post-processing\*\*: Implement post-processing steps like thresholding to make the final prediction (e.g., deciding whether an article is real or fake) based on the model's output probabilities.

11. \*\*User Interface\*\*: If this system is intended for public use, consider developing a user-friendly interface where users can submit articles for analysis and receive results.

12. \*\*Regular Updates\*\*: Fake news detection models should be regularly updated to adapt to new tactics used by malicious actors to spread misinformation.

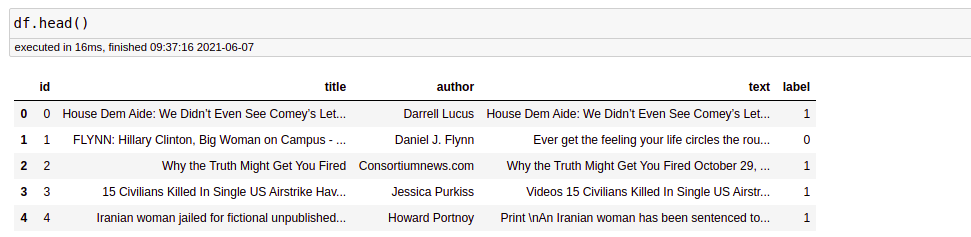
13. \*\*Ethical Considerations\*\*: Ensure that the fake news detection system is designed with ethical considerations in mind, avoiding biases and privacy violations.

Remember that fake news detection is a challenging and ongoing problem, and the effectiveness of NLP models in detecting fake news can vary. Continuous research and development are essential to stay ahead of those who create and spread false information.

df=pd.read\_csv('fake-news/train.csv')

df.head()

**output:-**



Before proceeding, we need to check whether a null value is present in our dataset or not.

df.isnull().sum()

There is no null value in this dataset. But if you have null values present in your dataset then you can fill it. In the code given below, I will tell you how you can replace the null values.

df = df.fillna(' ')

**3.BUILD LOADING AND PREPROCESSING THE DATASET**

To load and preprocess a dataset for fake news detection, you can follow these general steps:

Dataset Collection: First, you need a dataset containing both real and fake news articles. Datasets like the "Fake News Challenge" dataset or "LIAR" dataset are common choices.

Data Cleaning: This step involves removing any irrelevant information, special characters, HTML tags, or other noise from the text data.

Tokenization: Tokenize the text, splitting it into words or subword tokens. You can use libraries like NLTK, spaCy, or the Hugging Face Transformers library.

Stopword Removal: Remove common stopwords (e.g., "the," "is," "in") from the text, as they don't usually carry significant information.

Text Vectorization: Convert the text data into numerical vectors. Common methods include TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec, GloVe, or BERT embeddings.

Data Splitting: Split the dataset into training, validation, and test sets to train and evaluate your model.

Model-Specific Preprocessing: Depending on the model you plan to use (e.g., traditional machine learning or deep learning), you might need to apply specific preprocessing steps. For deep learning models, padding sequences to a fixed length is common.

Label Encoding: Encode the target labels (real or fake) into numerical values, typically 0 and 1.

Data Loader: Create data loaders or generators to efficiently load and batch the dataduring training.

Data Augmentation (Optional): For text data, data augmentation techniques are limited but can still be useful. Techniques like synonym replacement or back-translation can help generate additional training examples.

Normalization (Optional): For deep learning models, you might need to normalize the input data, especially when using models like BERT.

Final Preprocessing Checks: Ensure that all data is in the right format for your chosen model.

Here's an example code snippet in Python using libraries like pandas and scikit-learn:

python

Copy code

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

# Load the dataset

data = pd.read\_csv('fake\_news\_dataset.csv')

# Data Cleaning and Preprocessing (Steps 2-6)

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['text'], data['label'], test\_size=0.2, random\_state=42)

# Text Vectorization (Step 5)

tfidf\_vectorizer = TfidfVectorizer(max\_features=5000)

X\_train\_tfidf = tfidf\_vectorizer.fit\_transform(X\_train)

X\_test\_tfidf = tfidf\_vectorizer.transform(X\_test)

# Label Encoding (Step 8)

label\_encoder = LabelEncoder()

y\_train\_encoded = label\_encoder.fit\_transform(y\_train)

y\_test\_encoded = label\_encoder.transform(y\_test)

These are general guidelines, and the specific preprocessing steps may vary depending on your dataset and the machine learning or deep learning model you intend to use for fake news detection.

**PROGRAM:**

Here is an example of how to load and preprocess the dataset using Python:

Python

import pandas as pd

# Load the dataset

df = pd.read\_csv('fake\_news\_dataset.csv')

# Preprocess the text data

def preprocess\_text(text):

# Lowercase the text

text = text.lower()

# Remove punctuation and digits

text = re.sub('[^\w\s]', '', text)

# Remove stop words

stopwords = nltk.corpus.stopwords.words('english')

text = ' '.join([word for word in text.split() if word not in stopwords])

# Stem or lemmatize the text

# (This step is optional, but it can improve the performance of the model)

return text

# Apply the preprocessing function to the text column

df['text'] = df['text'].apply(preprocess\_text)

# Split the dataset into training and test sets

from sklearn.model\_selection import train\_test\_split

X = df['text']

y = df['label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25)

# Save the preprocessed data

X\_train.to\_csv('X\_train.csv', index=False)

X\_test.to\_csv('X\_test.csv', index=False)

y\_train.to\_csv('y\_train.csv', index=False)

y\_test.to\_csv('y\_test.csv', index=False

**Predicting the Model:**

In [22]:

linkcode

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df.clean\_joined\_text, df.target, test\_size = 0.2,random\_state=2)

vec\_train = CountVectorizer().fit(X\_train)

X\_vec\_train = vec\_train.transform(X\_train)

X\_vec\_test = vec\_train.transform(X\_test)

model = LogisticRegression(C=2.5)

model.fit(X\_vec\_train, y\_train)

predicted\_value = model.predict(X\_vec\_test)

accuracy\_value = roc\_auc\_score(y\_test, predicted\_value)

print(accuracy\_value)

**OUTPUT:**

0.9953661308915527

prediction = []

for i **in** range(len(predicted\_value)):

if predicted\_value[i].item() > 0.5:

prediction.append(1)

else:

prediction.append(0)

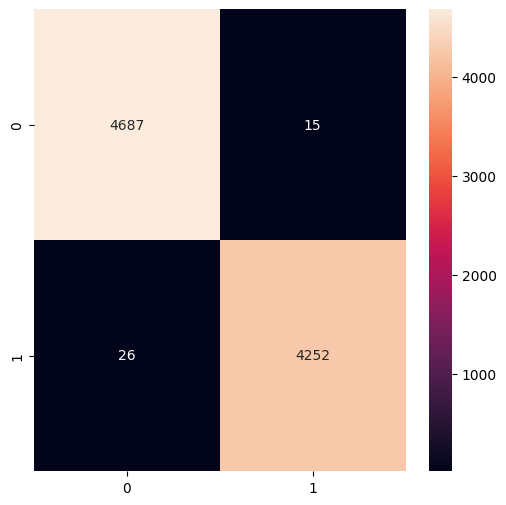
cm = confusion\_matrix(list(y\_test), prediction)

plt.figure(figsize = (6, 6))

sns.heatmap(cm, annot = True,fmt='g')

Out[23]:

<Axes: >



**Creating Prediction Model:**

In [16]:

linkcode

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df.clean\_joined\_title, df.target, test\_size = 0.2,random\_state=2)

vec\_train = CountVectorizer().fit(X\_train)

X\_vec\_train = vec\_train.transform(X\_train)

X\_vec\_test = vec\_train.transform(X\_test)

**Create the confusion matrix:**

In [18]:

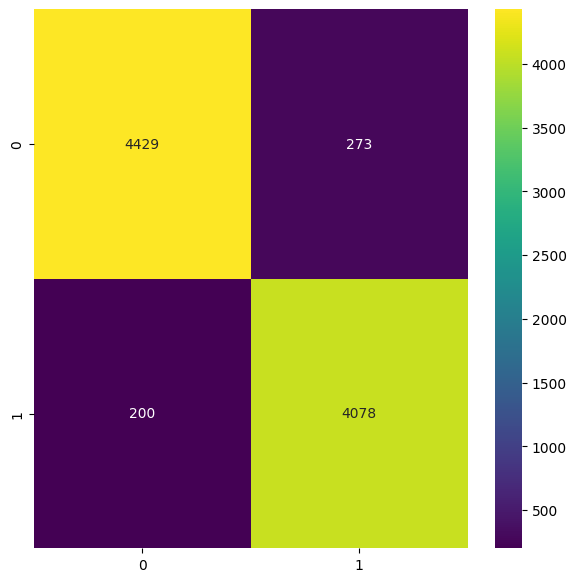
cm = confusion\_matrix(list(y\_test), predicted\_value)

plt.figure(figsize = (7, 7))

sns.heatmap(cm, annot = True,fmt='g',cmap='viridis')

Out[18]:

<Axes: >



*# Passive Aggresive Classifier*

pac = PassiveAggressiveClassifier(max\_iter=50)

pac.fit(tfidf\_train,y\_train)

pred = pac.predict(tfidf\_test)

print("Accuracy score : **{}**".format(accuracy\_score(y\_test, pred)))

print("Confusion matrix : **\n** **{}**".format(confusion\_matrix(y\_test, pred)))

Accuracy score : 0.936069455406472

Confusion matrix :

[[592 38]

[ 43 594]]

In [22]:

*# Logistic Regression model*

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression(max\_iter = 500)

lr.fit(tfidf\_train, y\_train)

print('Logistic Regression model fitted..')

pred = lr.predict(tfidf\_test)

print("Accuracy score : **{}**".format(accuracy\_score(y\_test, pred)))

print("Confusion matrix : **\n** **{}**".format(confusion\_matrix(y\_test, pred)))

Logistic Regression model fitted..

Accuracy score : 0.9179163378058406

Confusion matrix :

[[565 65]

[ 39 598]]

import xgboost

from xgboost import XGBClassifier

xgb = XGBClassifier()

xgb.fit(tfidf\_train, y\_train)

print('XGBoost Classifier model fitted..')

pred = xgb.predict(tfidf\_test)

print("Accuracy score : **{}**".format(accuracy\_score(y\_test, pred)))

print("Confusion matrix : **\n** **{}**".format(confusion\_matrix(y\_test, pred)))

XGBoost Classifier model fitted..

Accuracy score : 0.9289660615627466

Confusion matrix :

[[587 43]

[ 47 590]]

In [24]:

linkcode

import lightgbm

from lightgbm import LGBMClassifier

lgbm.fit(tfidf\_train, y\_train)

print('LightGBM Classifier model fitted..')

pred = lgbm.predict(tfidf\_test)

print("Accuracy score : **{}**".format(accuracy\_score(y\_test, pred)))

print("Confusion matrix : **\n** **{}**".format(confusion\_matrix(y\_test, pred)))

LightGBM Classifier model fitted..

Accuracy score : 0.9289660615627466

Confusion matrix :

[[581 49]

[ 41 596]]

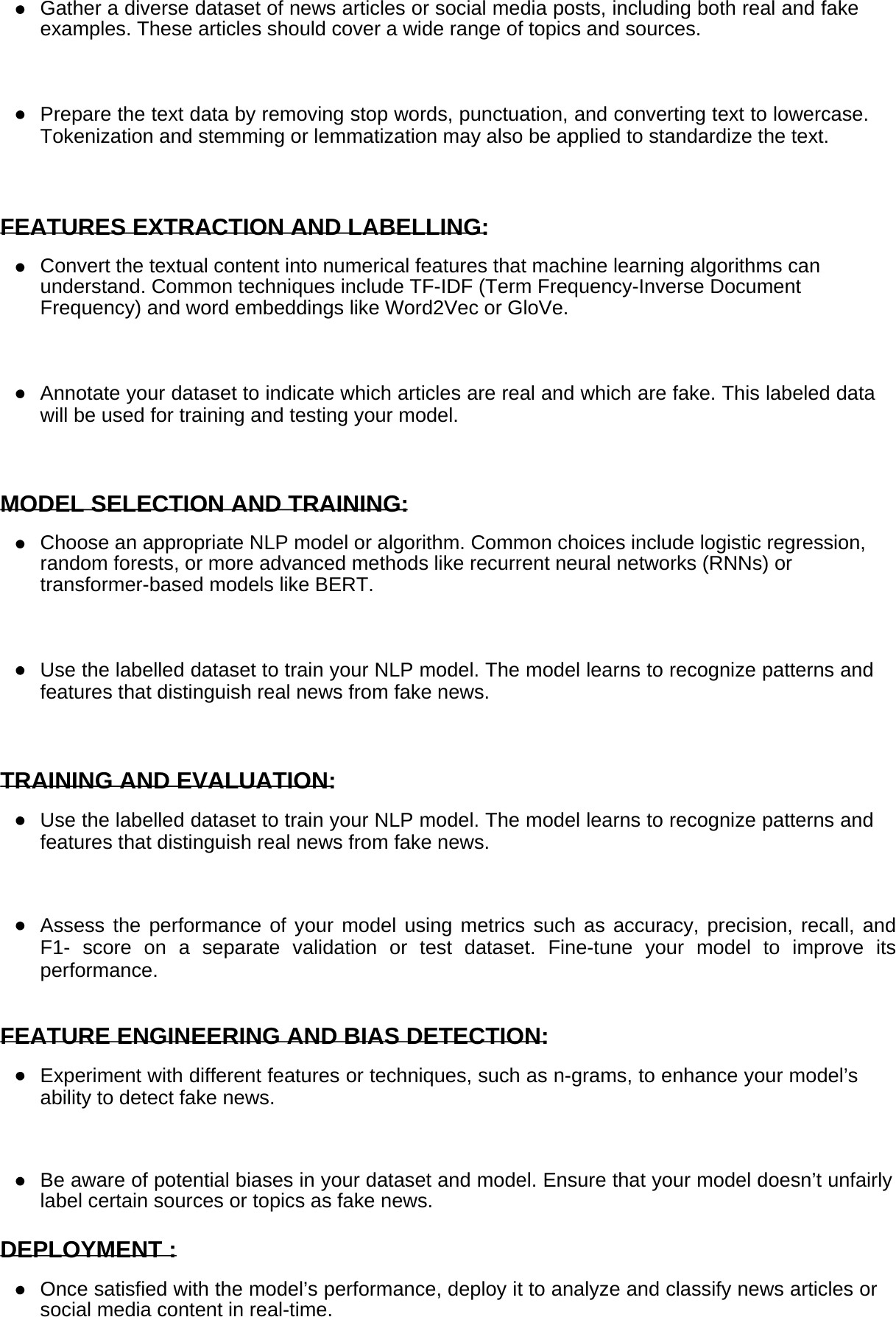
**4.PERFORMING DIFFERENT ACTIVITIES LIKE FEATURE ENGINEERING,MODEL TRAINING EVALUATION etc**

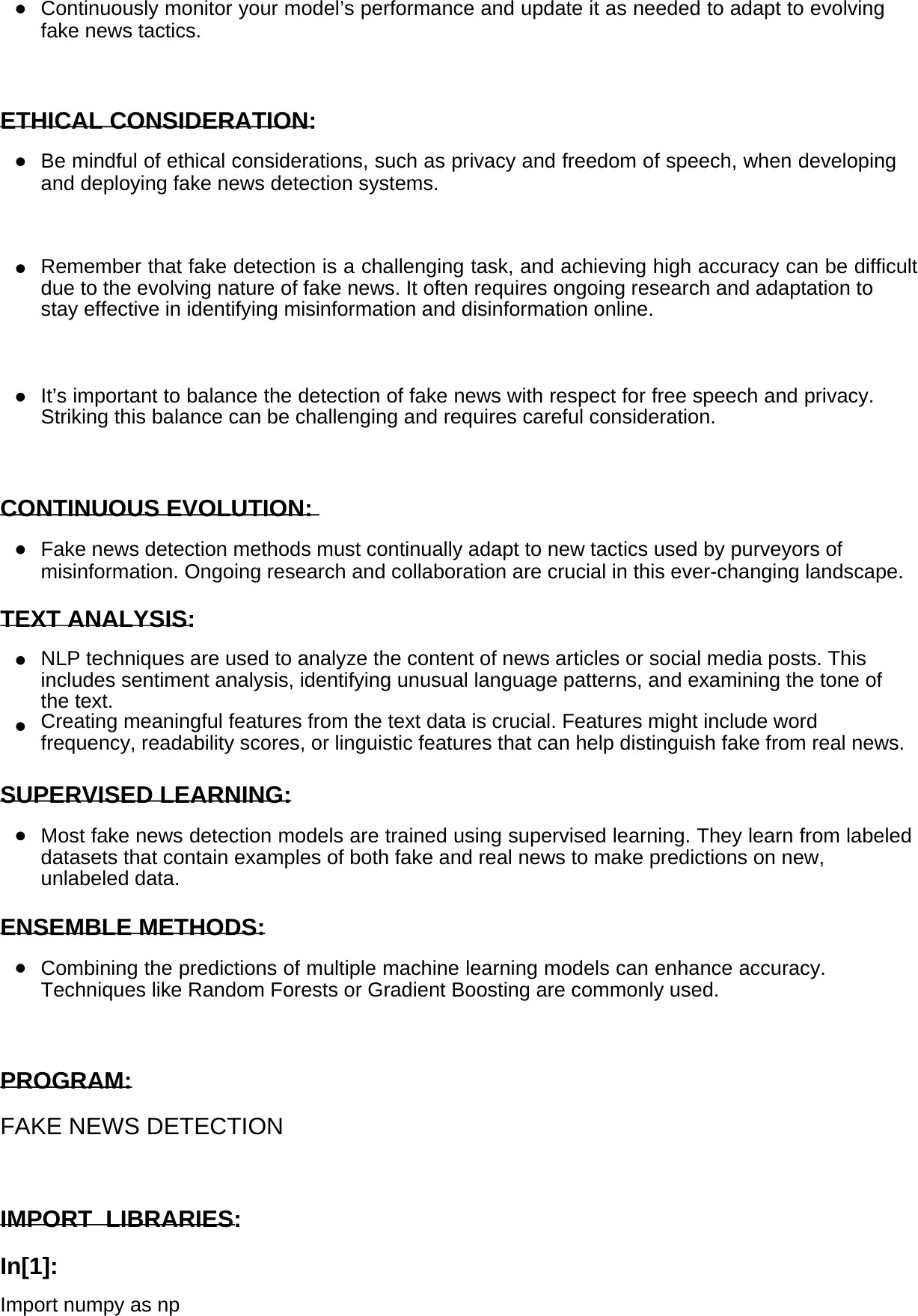
****

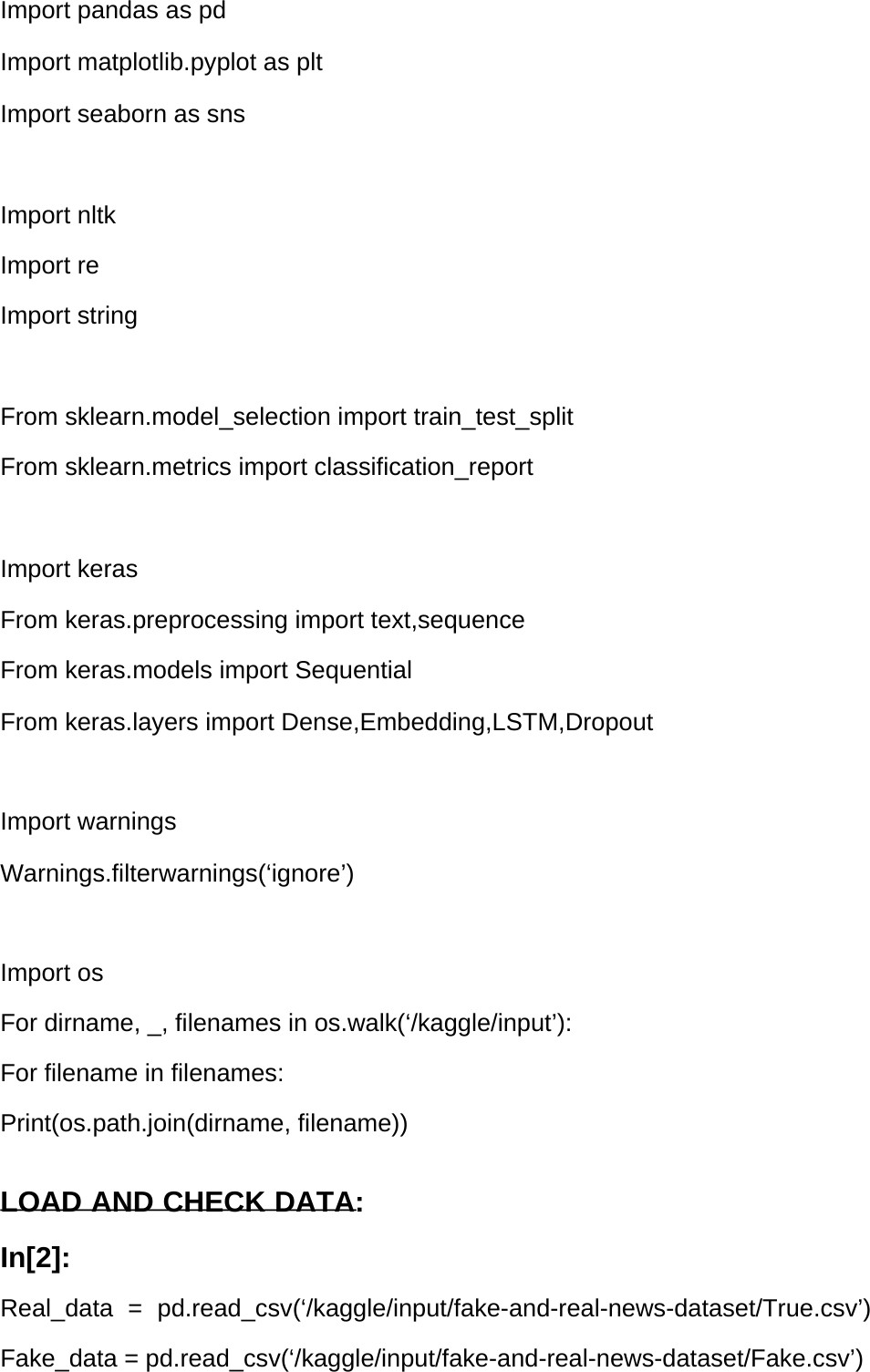
Motivations for Fake News: Fake news can be created for various reasons, such as political manipulation, financial gain, or simply for entertainment. It often seeks to exploit emotions, biases, or controversy to gain attention and traction.

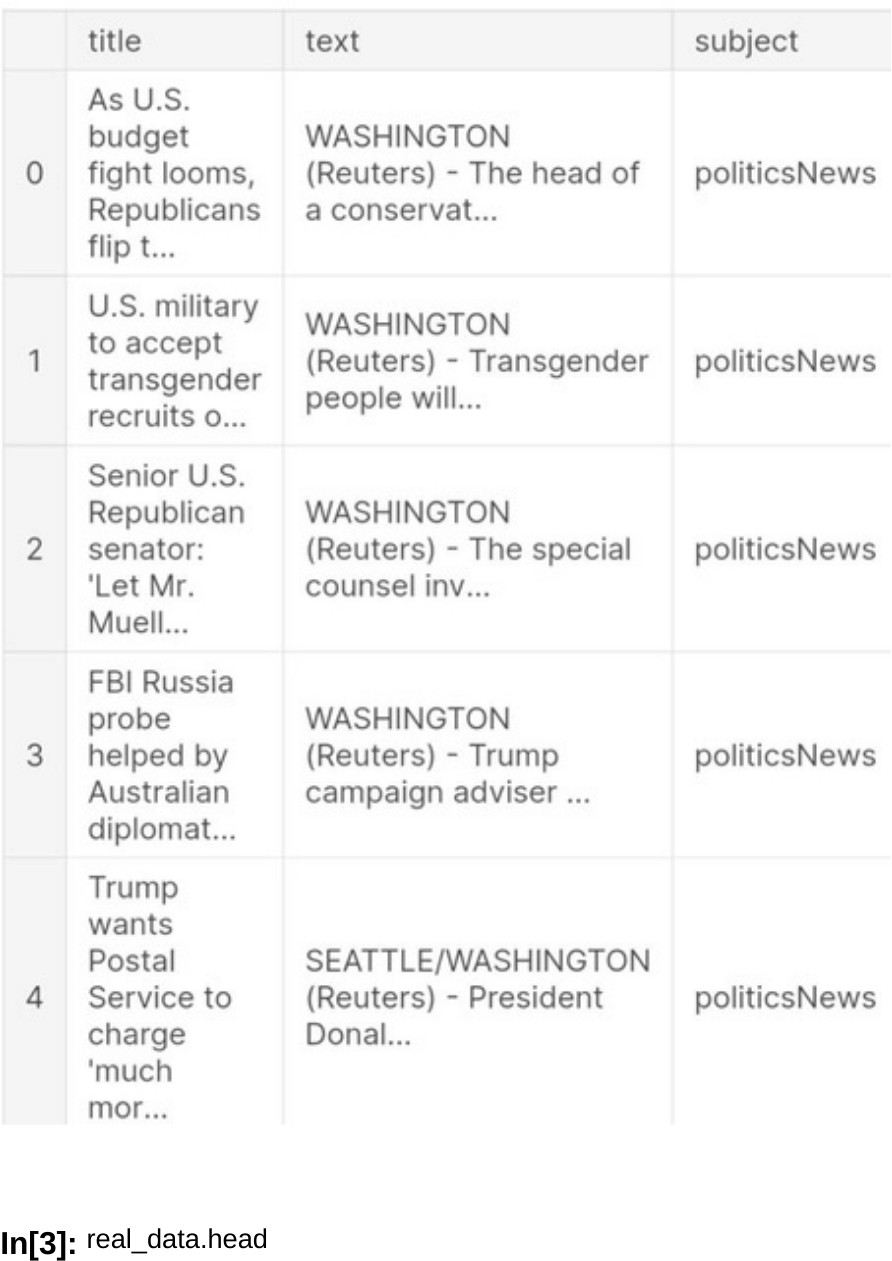
Impact of Fake News: Fake news can have serious consequences, including influencing public opinion, swaying elections, causing panic, or harming individuals' reputations. It can erode trust

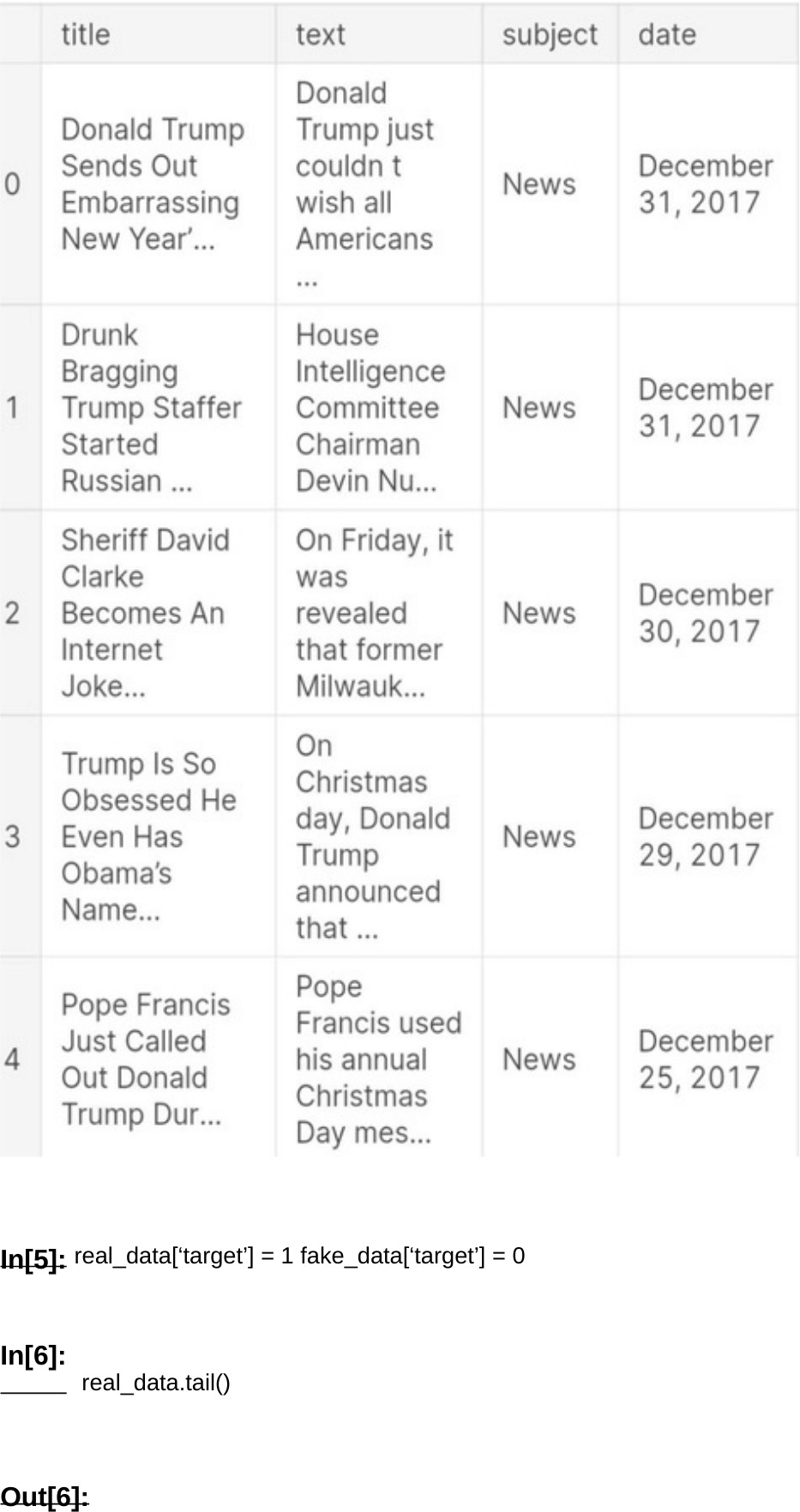




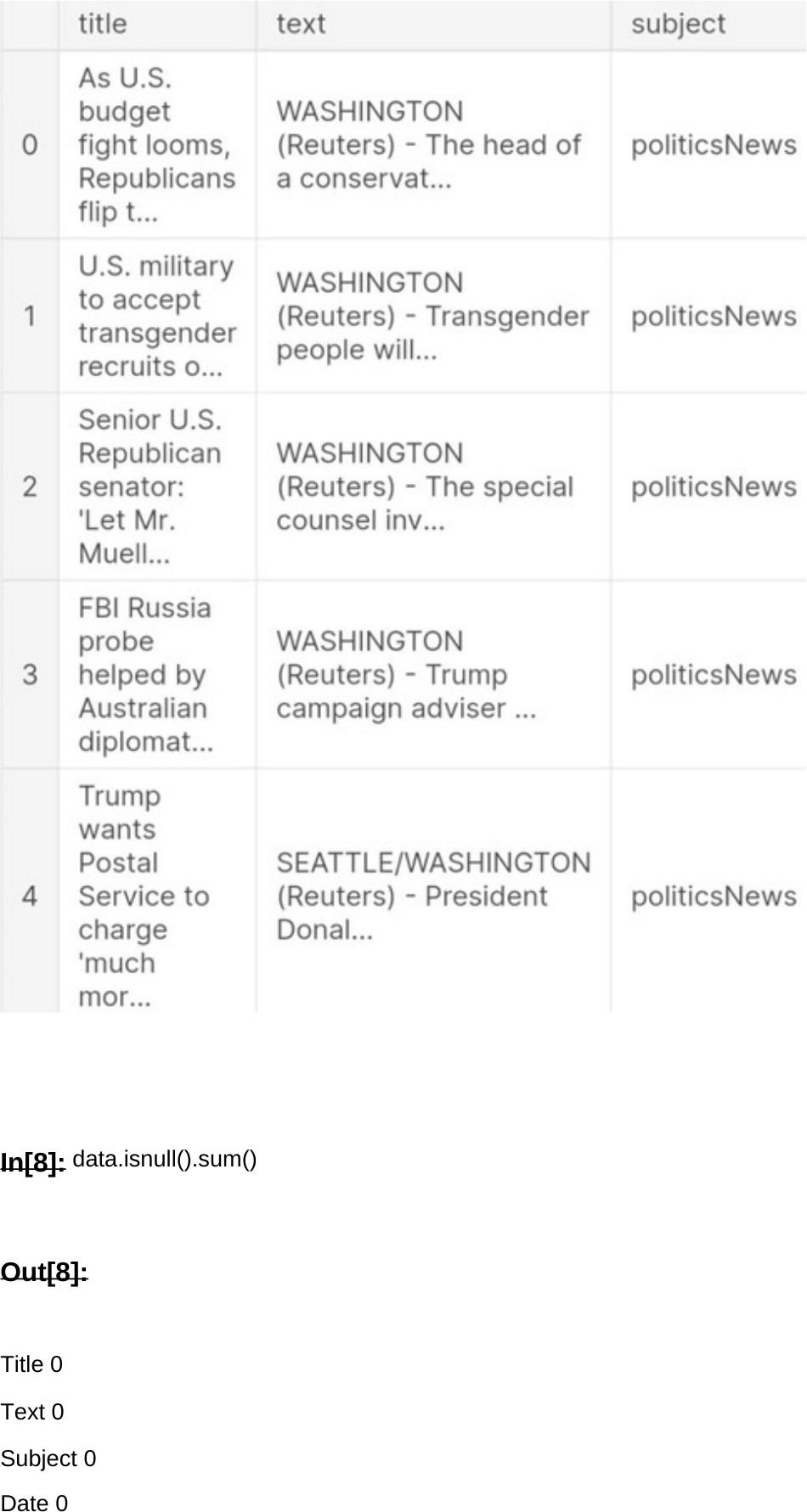


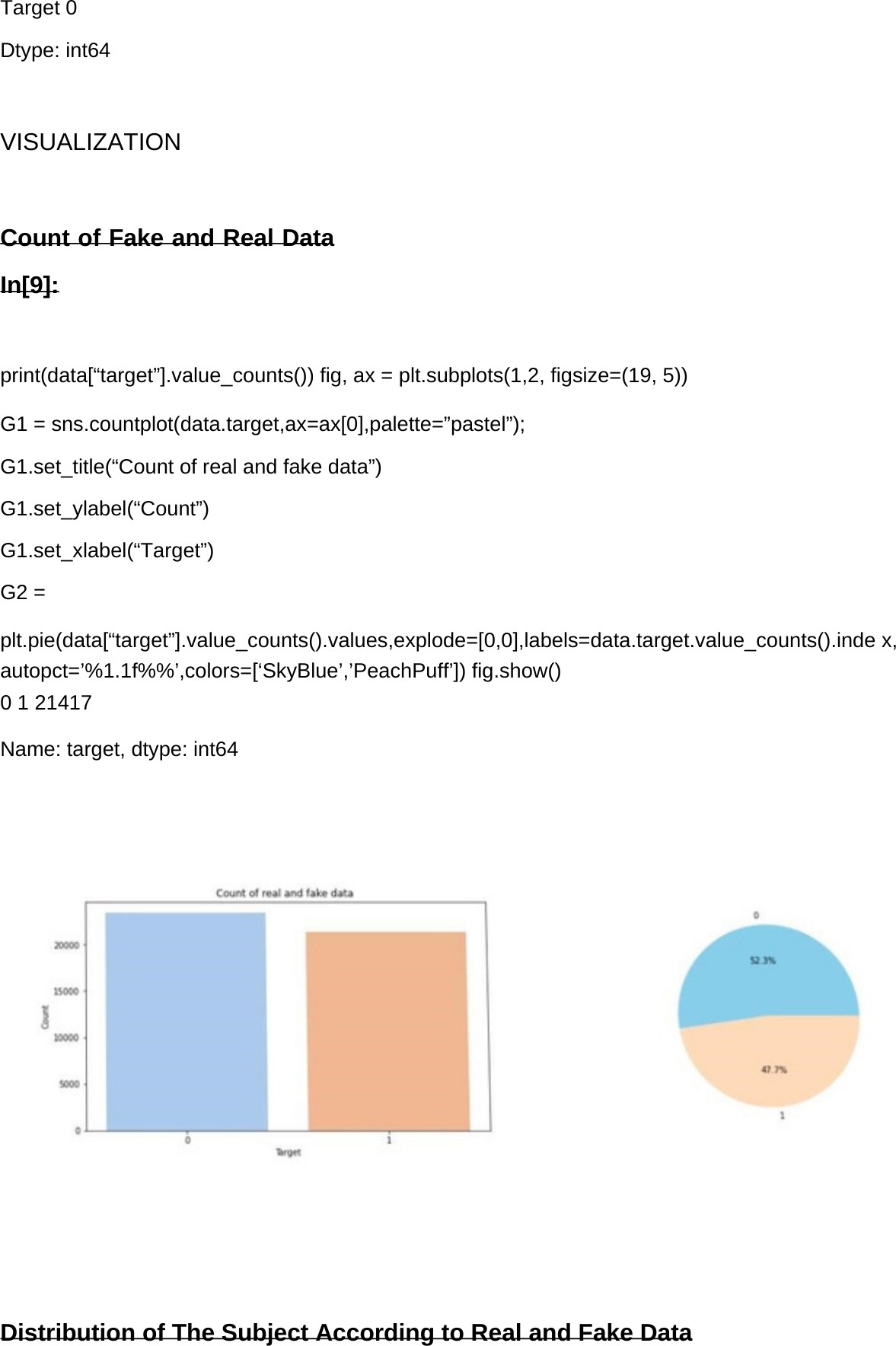


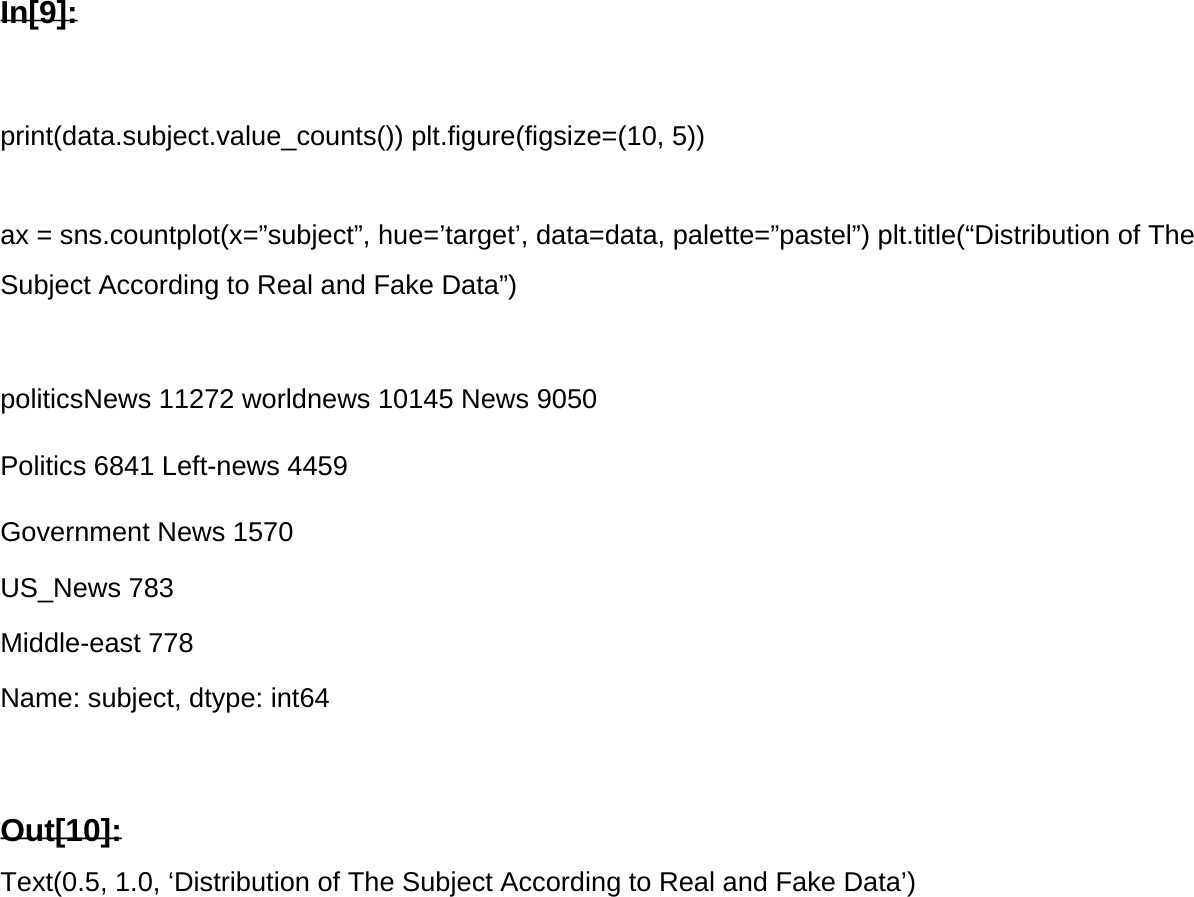




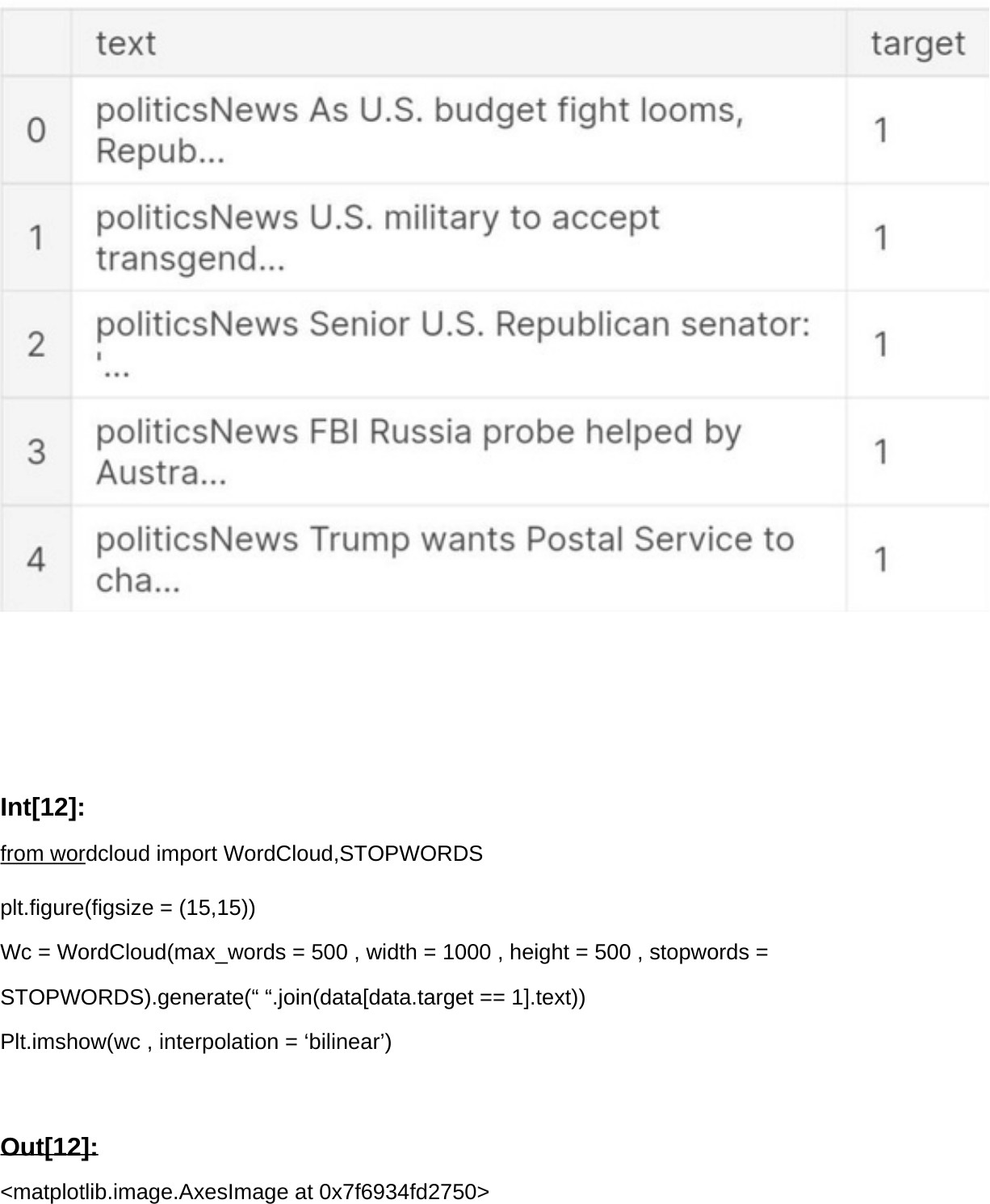




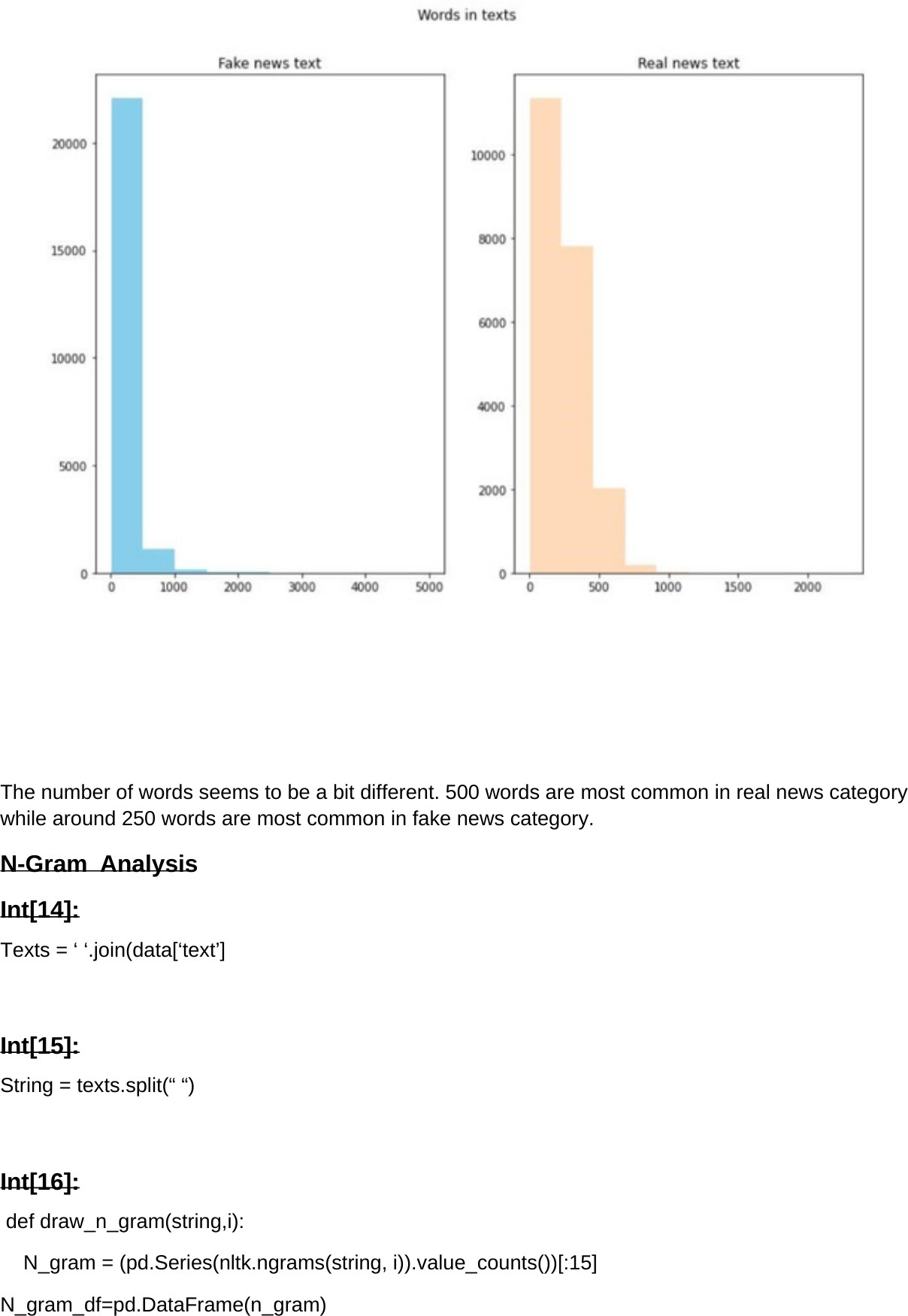


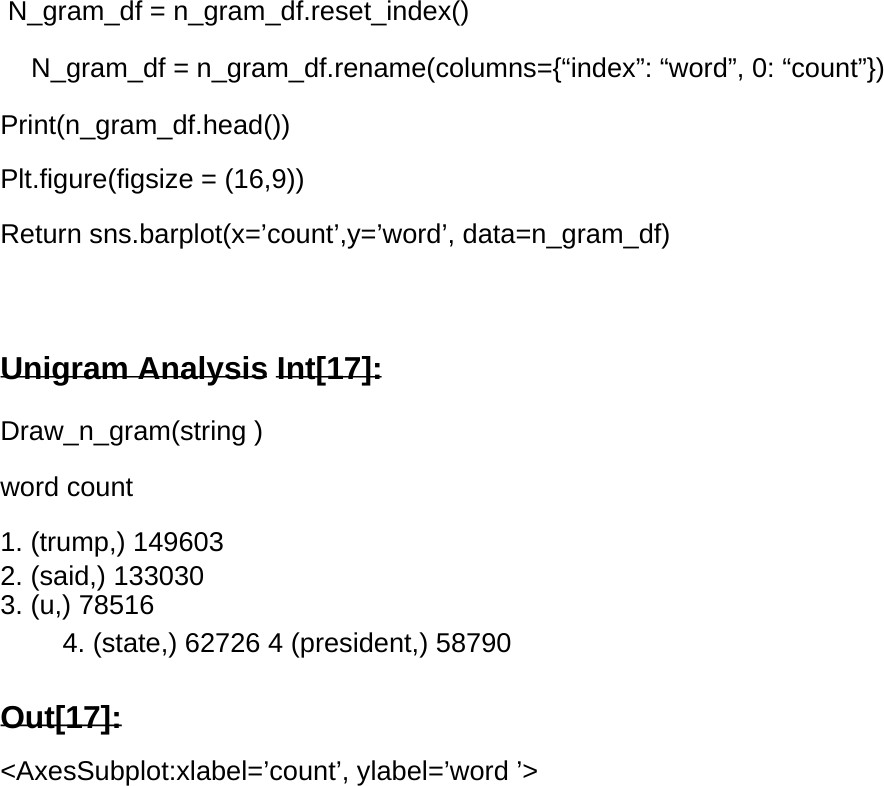


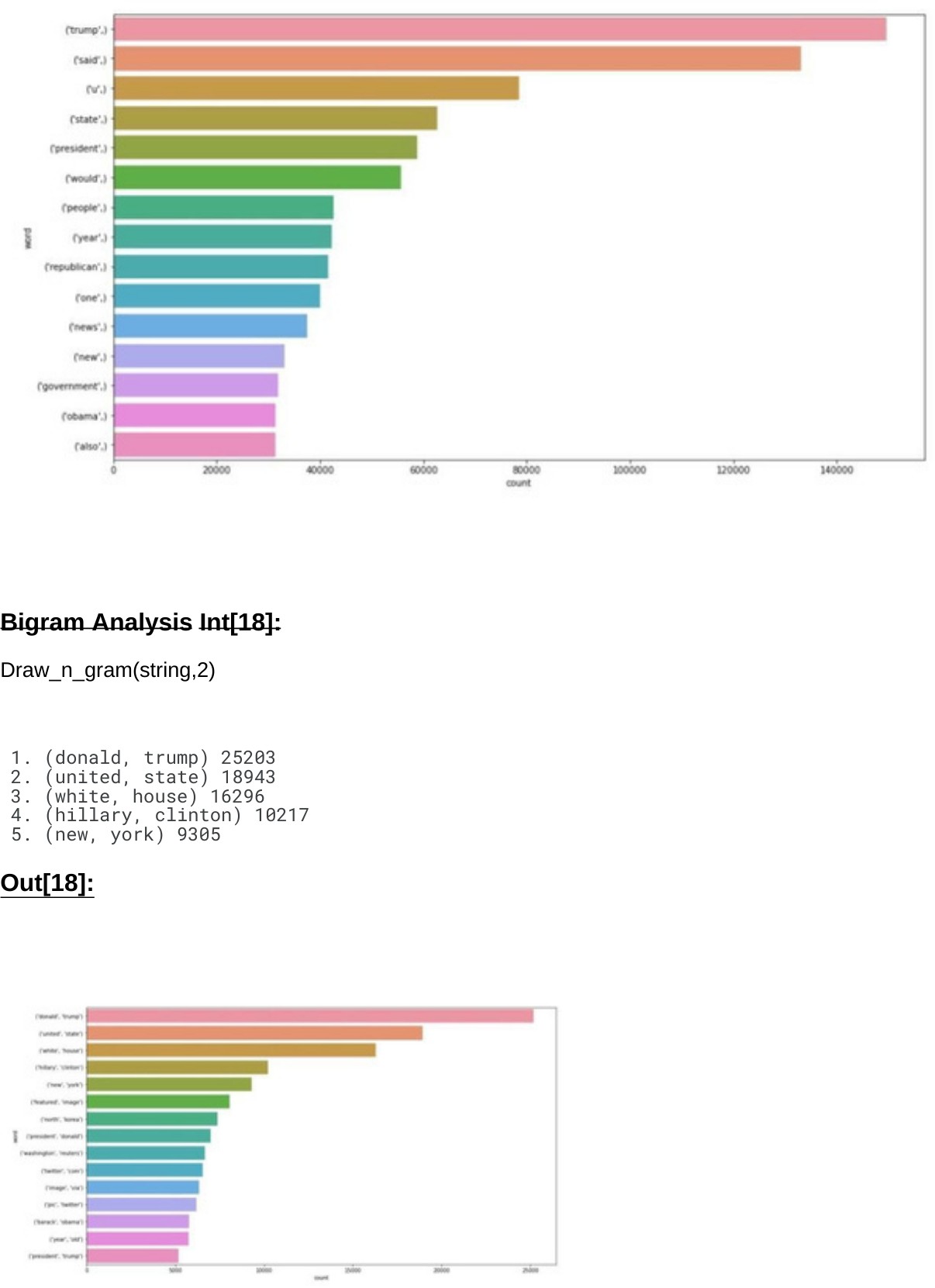


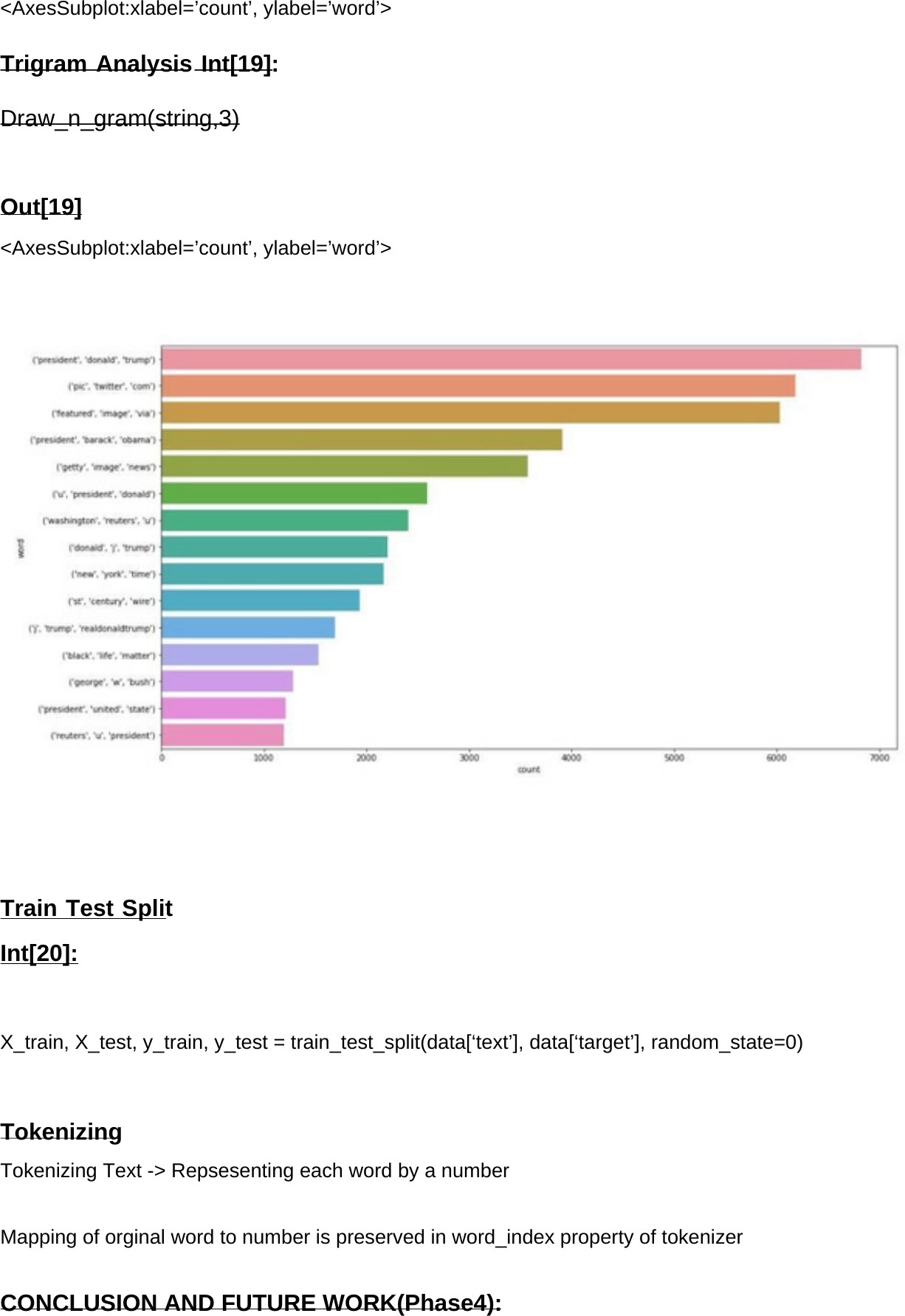












**CONCLUSION:**

Fake news have increased in recent years and it has caused a lot of harm to the society. This research project aimed to develop a model using the techniques of NLP and ML to detect if a news article/headline is fake or not and identify which methods give better output. In this paper, we have presented six LSTM models and three different methods were used for feature extraction. We have used different attributes like the title and text of the news to perform fake news detection. For future work we can work on larger dataset and also future research can be done on images , videos which can help in improving the models. The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal Productions from the IEEE LaTeX style files. Causal Productions offers both A4 templates and US Letter templates for LaTeX and Microsoft Word. The LaTeX templates depend on the official IEEEtran.cls and IEEEtran.bst files, whereas the Microsoft Word templates are self-contained. Causal Productions has used its best efforts to ensure that the templates have the same appearance. Causal Productions permits the distribution and revision of these templates on the condition that Causal Productions is credited in the revised template as follows: “original version of this template was provided by courtesy of Causal Productions.